

AMENDMENT(S) TO THE CLAIMS

1
2
3
4 1. (currently amended) One or more processor-accessible storage
5 media comprising processor-executable instructions that, when executed, direct a
6 device to perform actions comprising:

7 determining if an instruction of a line of common intermediate language
8 code meets a predetermined exception-related criterion; and

9 if so, injecting a decision point in association with the instruction of the line
10 of common intermediate language code, the decision point enabling a decision as
11 to whether an exception is to be thrown with respect to the instruction; wherein the
12 action of injecting a decision point comprises an action of injecting a bookmark
13 entry, the bookmark entry including an instruction type indicator that indicates an
14 instruction type for the instruction of the line of common intermediate language
15 code.
16
17
18
19
20
21
22
23
24
25

1 2. (currently amended) The one or more processor-accessible storage
2 media as recited in claim 1, comprising the processor-executable instructions that,
3 when executed, direct the device to perform further actions comprising:

4 retrieving the line of common intermediate language code from a common
5 intermediate language code program prior to the determining;

6 retrieving another line of common intermediate language code from the
7 common intermediate language code program; and

8 repeating the action of determining and the action of injecting a decision
9 point for an instruction of the ~~other~~-retrieved other line of common intermediate
10 language code.

11
12 3. (currently amended) The one or more processor-accessible storage
13 media as recited in claim 1, wherein the action of determining comprises an action
14 of:

15 determining if the instruction of the line of common intermediate
16 language code is capable of throwing an exception.

17
18 4. (currently amended) The one or more processor-accessible storage
19 media as recited in claim 1, wherein the action of determining comprises an action
20 of:

21 determining if the instruction of the line of common intermediate
22 language code is capable of throwing an exception and is related to a pre-
23 selected exception area.
24
25

1 5. (currently amended) The one or more processor-accessible storage
2 media as recited in claim 1, wherein the action of determining comprises an action
3 of:

4 determining if the instruction of the line of common intermediate
5 language code is capable of throwing an exception with reference to a
6 common intermediate language code specification.

7
8 6. (canceled)

9
10 7. (canceled)

11
12 8. (currently amended) The one or more processor-accessible storage
13 media as recited in claim ~~6~~ 1, wherein the ~~action of injecting a bookmark entry~~
14 ~~comprises an action of:~~

15 ~~injecting further includes~~ an identifier that uniquely identifies
16 the decision point within the common intermediate language code
17 that is being instrumented.

18
19 9. (currently amended) The one or more processor-accessible storage
20 media as recited in claim 1, wherein the action of injecting a decision point further
21 comprises an action of:

22 injecting a call to a decision runtime library, the decision runtime
23 library comprising a program that is adapted to evaluate whether the
24 exception is to be thrown with respect to the instruction.

10. (currently amended) The one or more processor-accessible storage
media as recited in claim 1, wherein at least a portion of the processor-executable
instructions comprise at least part of an instrumentation tool that produces
instrumented common intermediate language code from the common intermediate
language code by repeating the actions of determining and injecting a decision
point for a plurality of respective instructions of a plurality of respective lines of
the common intermediate language code.

11. (canceled)

12. (currently amended) A device comprising:
instrumented common intermediate language code that includes a test
couplet corresponding to a decision point and an associated instruction, the
associated instruction capable of causing a fault;

a decision runtime library that is adapted to evaluate the test couplet to
selectively decide whether to throw an exception with respect to the associated
instruction; and

a common language runtime component that interprets the decision point so
as to call the decision runtime library prior to executing the associated instruction;

wherein the decision point comprises a bookmark entry and a call to the
decision runtime library, and wherein the bookmark entry comprises an indication
of an instruction type of the associated instruction and an identifier of the decision
point.

1 13. (original) The device as recited in claim 12, wherein the
2 instrumented common intermediate language code is in a binary form.

3
4 14. (canceled)

5
6 15. (currently amended) The device as recited in claim ~~14~~ 12, wherein
7 the decision runtime library is further adapted to evaluate the test couplet to
8 selectively decide whether to throw an exception responsive to the bookmark
9 entry.

10
11 16. (currently amended) The device as recited in claim ~~14~~ 12, wherein
12 the decision runtime library is further adapted to evaluate the test couplet to
13 selectively decide whether to throw an exception responsive to the bookmark entry
14 and based on throw exception decision logic.

15
16 17. (currently amended) The device as recited in claim ~~14~~ 12, wherein
17 the decision runtime library is further adapted to evaluate the test couplet to
18 selectively decide whether to throw an exception responsive to the bookmark entry
19 and based on at least one throw exception decision logic factor selected from the
20 group comprising: throwing an exception randomly, throwing an exception when
21 first encountering a given decision point using an identifier of the given decision
22 point, and throwing an exception when encountering a particular decision point
23 along each new execution path using an identifier of the particular decision point
24 and one or more stack values.
25

1 **18.** (currently amended) The device as recited in claim ~~14~~ 12, wherein
2 the decision runtime library is further adapted to evaluate the test couplet to
3 selectively decide whether to throw an exception responsive to the indication of
4 the instruction type of the associated instruction.

5
6 **19.** (currently amended) The device as recited in claim 12, wherein the
7 decision runtime library is (i) modularized by exception category and/or (ii)
8 operative in dependence on an instruction type of the associated instruction as
9 determinable by the indication of the instruction type from the bookmark entry.

1 20. (currently amended) An arrangement for enabling reliability testing
2 of managed code, the arrangement including one or more processor-accessible
3 storage media; wherein the arrangement comprisescomprising:

4 instrumentation means for instrumenting common intermediate language
5 code with a plurality of decision points to produce instrumented common
6 intermediate language code;

7 wherein the instrumentation means comprises:

8 analysis means for analyzing whether individual instructions of a
9 plurality of instructions of the common intermediate language code can
10 result in a failure; and

11 injection means for injecting a respective decision point in
12 association with each respective individual instruction, which can result in a
13 failure as analyzed by the analysis means, of the plurality of instructions of
14 the common intermediate language code;

15 wherein the injection means comprises:

16 means for injecting a respective bookmark entry that indicates
17 an instruction type of the respective individual instruction associated
18 with the respective decision point; and

19 decision means for deciding whether to throw an exception at each decision
20 point of the plurality of decision points.
21
22
23
24
25

1 21. (currently amended) The arrangement as recited in claim 20,
2 wherein the instrumentation means comprises: respective bookmark entry further
3 identifies the respective decision point.

4 ~~analysis means for analyzing whether individual instructions of a~~
5 ~~plurality of instructions of the common intermediate language code can~~
6 ~~result in a failure; and~~

7 ~~injection means for injecting a respective decision point in~~
8 ~~association with each respective individual instruction, which can result in a~~
9 ~~failure as analyzed by the analysis means, of the plurality of instructions of~~
10 ~~the common intermediate language code.~~

11
12 22. (currently amended) The arrangement as recited in claim ~~21~~ 20,
13 wherein the injection means further comprises:

14 ~~means for injecting a respective bookmark entry that indicates~~
15 ~~an instruction type of the respective individual instruction associated~~
16 ~~with the respective decision point and that identifies the respective~~
17 ~~decision point; and~~

18 ~~means for injecting a call at least one module that is capable~~
19 ~~of evaluating the respective decision point with regard to whether a~~
20 ~~failure is to be induced.~~

1 23. (original) The arrangement as recited in claim 20, further
2 comprising:

3 common language runtime means for executing the instrumented common
4 intermediate language code and the decision means in a runtime environment.

5
6 24. (original) The arrangement as recited in claim 23, wherein the
7 decision means operates while the instrumented common intermediate language
8 code is being executed when the common language runtime means calls the
9 decision means at each decision point of the plurality of decision points.

10
11 25. (currently amended) The arrangement as recited in claim 20,
12 wherein the decision means comprises:

13 evaluation means for evaluating whether to throw an exception
14 responsive to ~~a~~ the respective bookmark entry of each respective decision
15 point of the plurality of decision points and based on at least one throw
16 exception decision logic factor.

17
18 26. (currently amended) The arrangement as recited in claim 20,
19 wherein the arrangement comprises at least one device having the one or more
20 processor-accessible storage media.

21
22 27. (canceled)
23
24
25

1 28. (currently amended) One or more processor-accessible storage
2 media comprising an instrumentation tool that is capable of determining whether
3 respective instructions from common intermediate language code meet at least one
4 predetermined exception-related criterion and that is adapted to inject respective
5 decision points into the common intermediate language code in association with
6 the respective instructions that meet the at least one predetermined exception-
7 related criterion, each injected respective decision point including an indication of
8 an instruction type of the respective associated instruction, an identifier of the
9 injected respective decision point, and a call to a program that can selectively
10 cause an exception to be thrown with respect to the respective associated
11 instruction.

12
13 29. (currently amended) A method for instrumentation injection with
14 regard to a common language runtime environment, the method comprising:

15 determining whether an instruction from common intermediate language
16 code is capable of causing an exception; and

17 if so, injecting a decision point in association with the instruction to mark
18 the instruction for evaluation during a common language runtime execution, the
19 evaluation involving a decision as to whether a failure is to be induced with
20 respect to the instruction; wherein the injecting a decision point comprises
21 injecting an indicator of an instruction type of the instruction.
22
23
24
25

1 30. (original) The method as recited in claim 29, wherein the
2 determining comprises:

3 determining whether the instruction from the common intermediate
4 language code is capable of causing an exception and is (i) related to a pre-
5 selected exception category and/or (ii) of a pre-selected instruction type.

6
7 31. (currently amended) The method as recited in claim 29, wherein the
8 injecting a decision point further comprises:

9 ~~injecting an indicator of an instruction type of the instruction;~~ and
10 injecting an identifier of the decision point.

11
12 32. (currently amended) The method as recited in claim 29, wherein the
13 injecting a decision point further comprises:

14 injecting a call to at least one module of a decision runtime library
15 that is adapted to perform the evaluation.

16
17 33. (currently amended) One or more processor-accessible storage
18 media comprising processor-executable instructions that, when executed, direct a
19 device to perform the method as recited in claim 29.

1 34. (currently amended) The method as recited in claim 29, further
2 comprising:

3 repeating the determining and the injecting a decision point for a plurality
4 of instructions from the common intermediate language code; and

5 producing instrumented common intermediate language code as a result of
6 the repeating.

7
8 35. (original) The method as recited in claim 34, further comprising:

9 detecting the decision point in the instrumented common intermediate
10 language code during execution thereof; and

11 calling at least one module of a decision runtime library, which is adapted
12 to perform the evaluation, as a result of the detecting.

13
14 36. (original) The method as recited in claim 29, further comprising:

15 selectively deciding whether the execution is to fail at the decision point.

16
17 37. (original) The method as recited in claim 36, further comprising:

18 if it is decided at the selectively deciding that the execution is to fail at the
19 decision point, then choosing which exception of at least two exceptions is to be
20 thrown.

1 38. (original) The method as recited in claim 36, further comprising:
2 if it is decided at the selectively deciding that the execution is to fail at the
3 decision point, then inducing a failure in the execution of the common language
4 runtime with respect to the instruction.
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25